

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Kenneth Carbone et al.                      Art Unit : 2415  
Serial No. : 09/582,297                                      Examiner : Ramy M. Osman  
Filed : June 3, 2002                                          Conf. No. : 1661  
Title : ASYNCHRONOUS DATA PROTOCOL

**Mail Stop Appeal Brief - Patents**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

BRIEF ON APPEAL

Appellant hereby submits this Brief on Appeal in response to the final rejection dated August 7, 2009.

**TABLE OF CONTENTS**

<u><b>Section</b></u>	<u><b>Page(s)</b></u>
<b>I. Real Party in Interest.....</b>	<b>2</b>
<b>II. Related Appeals and Interferences.....</b>	<b>3</b>
<b>III. Status of Claims .....</b>	<b>4</b>
<b>IV. Status of Amendments .....</b>	<b>5</b>
<b>V. Summary of Claimed Subject Matter .....</b>	<b>6</b>
<b>VI. Grounds of Rejection .....</b>	<b>10</b>
<b>VII. Argument.....</b>	<b>11-14</b>
<b>VIII. Claims Appendix.....</b>	<b>15</b>
<b>IX. Evidence Appendix .....</b>	<b>25</b>
<b>X. Related Proceedings Appendix.....</b>	<b>26</b>

Applicant : Kenneth Carbone et al.  
Serial No. : 09/582,297  
Filed : June 3, 2002  
Page : 2 of 26

Attorney's Docket No.: 06975-0029006 / Personlization  
01-DIV3

**I. Real Party in Interest**

AOL LLC, the assignee of this application, is the real party in interest.

Applicant : Kenneth Carbone et al.  
Serial No. : 09/582,297  
Filed : June 3, 2002  
Page : 3 of 26

Attorney's Docket No.: 06975-0029006 / Personlization  
01-DIV3

## **II. Related Appeals and Interferences**

There are no related appeals or interferences.

Applicant : Kenneth Carbone et al.  
Serial No. : 09/582,297  
Filed : June 3, 2002  
Page : 4 of 26

Attorney's Docket No.: 06975-0029006 / Personlization  
01-DIV3

### **III. Status of Claims**

Claims 1-27, 37-60 are pending, with claims 1, 10, 17, 37, 45, and 51 being independent.

Claims 28-36, 61-69 have been cancelled.

Appellants have appealed the rejection of claims 1-27, 37-60.

Applicant : Kenneth Carbone et al.  
Serial No. : 09/582,297  
Filed : June 3, 2002  
Page : 5 of 26

Attorney's Docket No.: 06975-0029006 / Personlization  
01-DIV3

#### **IV. Status of Amendments**

None.

## **V. Summary of Claimed Subject Matter**

The following summarizes disclosure related to each independent claim with references to the application specification and drawings. The references to the specification and drawings are meant to be exemplary, and not limiting.

### **Independent claim 1**

Independent claim 1 is directed to a method of asynchronously transferring a plurality of data objects between client and host devices. See, e.g., page 6:5-9. The method comprises transmitting a request for a data transfer session from a client device to a host, the request identifying a plurality of data objects to be transferred between the client device and the host. See, e.g., Figs. 5a-b, step 132; Figs. 6a-b; page 8:1-4.; page 9:17-10:5. The method further comprises transmitting, in response to the received request, from the host to the client device a plurality of identifiers for data objects, wherein each identifier is assigned by the host and corresponds to a different one of the data objects to be transferred. See, e.g., Figs. 5a-b, step 144; Fig. 6a-b; page 9:7-9, page 10:6-18. The method further comprises transferring over a network between the host and client devices a data frame that includes an identifier and at least a portion of the corresponding data object. See, e.g., Figs. 8a-b, page 12:4-13:28. In addition, the method includes repeating the data frame transfers until the plurality of data objects have been transferred. See, e.g., Figs. 8a-b, page 12:4-13:28.

### **Independent claim 10**

Independent claim 10 is directed to a method of asynchronously transferring a plurality of data objects between client and host devices. See, e.g., page 6:5-9. The method comprises transmitting, from a client device to a host, a request for a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host. See, e.g., Figs. 5a-b, step 132; Figs. 6a-b; page 8:1-4.; page 9:17-10:5. The method also includes transmitting, in response to the received request, to a client device a plurality of identifiers and routings of one or more handling processes, wherein each identifier is assigned by the host and corresponds to one of the data objects identified in the request. See, e.g., Figs. 5a-b, step 144; Fig. 6a-b; page 9:7-9, page 10:6-18. The method also includes transferring between the client

and host devices a first data frame that includes a first identifier, a routing of a first handling process, and at least a portion of the data object corresponding to the first identifier. See, e.g., Figs. 8a-b, page 12:4-13:28. Additionally, the method comprises transferring between the client and host devices a second data frame that includes a second identifier, a routing of a second handling process, and at least a portion of the data object corresponding to the second identifier. See, e.g., Figs. 8a-b, page 12:4-13:28. The method also includes repeating the data frame transfers until the plurality of data objects have been transferred. See, e.g., Figs. 8a-b, page 12:4-13:28.

#### **Independent claim 17**

Independent claim 17 is directed to a method of asynchronously transferring data between host and client devices. See, e.g., page 6:5-9. The method comprises receiving from a client device a request a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host. See, e.g., Figs. 5a-b, step 132; Figs. 6a-b; page 8:1-4; page 9:17-10:5. The method also comprises sending to the client device a frame defining a session protocol that assigns an identifier to each data object identified in the request, wherein each identifier is assigned by the host. See, e.g., Figs. 5a-b, step 144; Fig. 6a-b; page 9:7-9, page 10:6-18. The method further comprises transferring a plurality of data frames between the client and host devices, each data frame comprising a data portion of a data object and an identifier assigned to the data object including said data portion. See, e.g., Figs. 8a-b, page 12:4-13:28.

#### **Independent claim 37**

Independent claim 37 is directed to a computer readable medium encoding a program of instructions to asynchronously transfer a plurality of data objects between client and host devices. See, e.g., page 6:5-9. The instructions comprise transmitting, from a client device to a host, a request for a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host. See, e.g., Figs. 5a-b, step 132; Figs. 6a-b; page 8:1-4; page 9:17-10:5. The instructions also comprise transmitting to a client device the plurality of identifiers for data objects, wherein each identifier is assigned by the host and corresponds to a different one of the data objects to be transferred. See, e.g., Figs. 8a-b, page

12:4-13:28. The instructions also include transferring over a network between the host and client devices a data frame that includes an identifier and at least a portion of the corresponding data object. See, e.g., Figs. 8a-b, page 12:4-13:28. Additionally, the instructions include repeating the data frame transfers until the plurality of data objects have been transferred. See, e.g., Figs. 8a-b, page 12:4-13:28.

#### **Independent claim 45**

Independent claim 45 is directed to a computer readable medium encoding a program of instructions to asynchronously transfer a plurality of data objects between client and host devices. See, e.g., page 6:5-9. The instructions comprise transmitting, from a client device to a host, a request for a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host. See, e.g., Figs. 5a-b, step 132; Figs. 6a-b; page 8:1-4.; page 9:17-10:5. The instructions further comprise transmitting to the client device a plurality of identifiers and routings of one or more handling processes, wherein each identifier is assigned by the host and corresponds to one of the data objects identified in the request. See, e.g., Figs. 5a-b, step 144; Fig. 6a-b; page 9:7-9, page 10:6-18. The instructions also include transferring between the client and host devices a first data frame that includes a first identifier, a routing of a first handling process, and at least a portion of the data object corresponding to the first identifier. See, e.g., Figs. 8a-b, page 12:4-13:28. Additionally, the instructions include transferring between the client and host devices a second data frame that includes a second identifier, a routing of a second handling process, and at least a portion of the data object corresponding to the second identifier. See, e.g., Figs. 8a-b, page 12:4-13:28. The instructions also include repeating the data frame transfers until the plurality of data objects have been transferred. See, e.g., Figs. 8a-b, page 12:4-13:28.

#### **Independent claim 51**

Independent claim 51 is directed to a computer readable medium encoding a program of instructions to asynchronously transferring data between host and client devices. See, e.g., page 6:5-9. The instructions include receiving from a client device a request for a data transfer session, the request identifying a plurality of data objects to be transferred between the client



device and the host. See, e.g., Figs. 5a-b, step 132; Figs. 6a-b; page 8:1-4.; page 9:17-10:5. The instructions also include sending to the client device a frame defining a session protocol that assigns an identifier to each data object identified in the request; wherein each identifier is assigned by the host. See, e.g., Figs. 5a-b, step 144; Fig. 6a-b; page 9:7-9, page 10:6-18. Further, the instructions include transferring a plurality of data frames between the client and host devices, each data frame comprising a data portion and an identifier assigned to a data object including said data portion. See, e.g., Figs. 8a-b, page 12:4-13:28.

## **VI. Grounds of Rejection**

A. Claims 1-27 and 37-60 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Bennett (6,963,923)

B. Claim 1 stands rejected under 35 U.S.C. § 112, second paragraph for failing to particularly point out the invention.

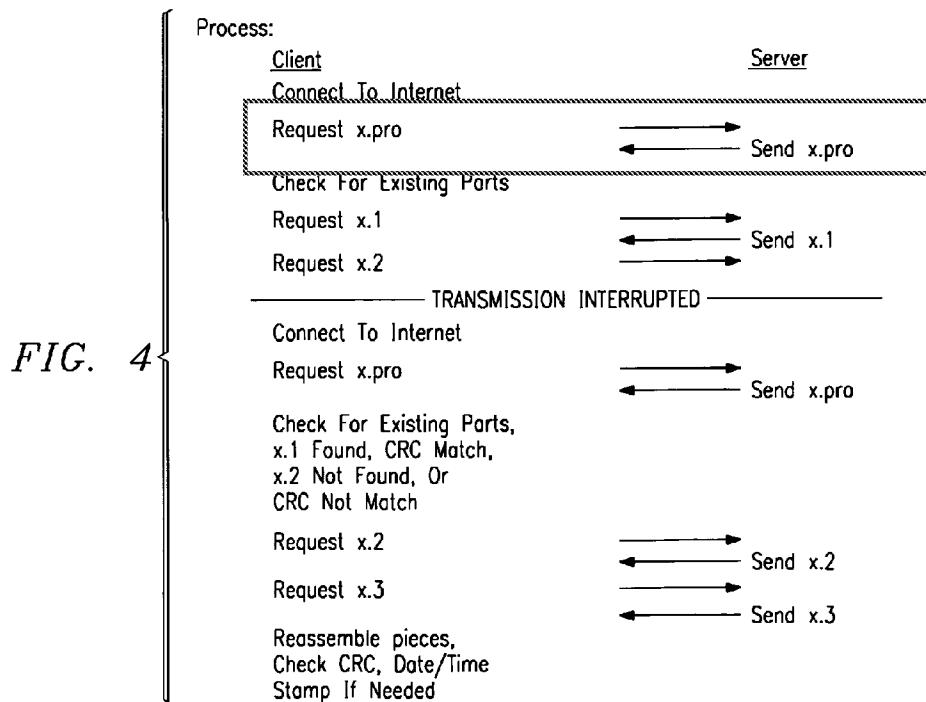
## VII. Argument

### A. Bennett Does Not Anticipate Claims 1-27 and 37-60.

Claims 1-27 and 37-60 stand rejected under 35 U.S.C. 102(e) as being anticipated by Bennett (6,963,923).

Claim 1 recites, among other features, “transmitting a request for a data transfer session from a client device to a host, the request identifying a plurality of data objects to be transferred between the client device and the host.” Bennett does not teach or suggest at least this feature.

Bennett describes a multi-component file transfer system, where each file is segmented into multiple components at the server. See, e.g., Abstract. To keep track of multiple components for each file, Bennett also stores their listing in the “profile” at the server. Id. Thus, each “profile” corresponds to a particular file and describes all components comprising that file. Id. First, a data session is established by a transmission of command files from a server to a client. Col. 8:47-50. Then, to initiate a download of file *x*, the client first sends a request for a “profile” of file *x* to the server. Fig 6, step 262; Col. 8:49-52. The server responds back with a “profile” *x.pro*, which identifies all individual components of the file *x* (i.e., *x.1*, *x.2*, *x.3*, etc) that the client needs to download from the server. Fig. 6, step 263; Col. 8:52-54. Once the client receives the “profile” for file *x* from the server, the client proceeds to individually and sequentially request each file component specified in the “profile” from the server. Fig. 6, step 266; Col. 8:65-9:1. In response to each client request for component *x.n*, the server responds by sending each component back to the client. Col. 9:2- 9:18. The entire download sequence is clearly shown in Figure 4 of Bennett, with the initial profile request and response being highlighted in red:



Critically, the initial client request in Bennett only identifies a single profile *x.pro*, but not the multiple objects within the profile. It is clear from Bennett that at the time of the first request, the client does not know which file components need to be downloaded from the server. Bennett explains that the actual profile identifying file components is transmitted from the server to the client, in response to the client's initial request.

Nevertheless, the Advisory Action asserts that the client's request for a profile *x.pro* in Bennett "is used, whether directly or indirectly, to identify a plurality of objects." Advisory Action of December 7, 2009 at 3. Applicants respectfully disagree.

In contrast to Bennett, claim 1 expressly indicates that a request that is transmitted from the client to the host must "identify[] a plurality of data objects". This language is unequivocal, calling for identification of the data objects by a single request from the client, not by a combination of multiple requests and not by something other than a request transmitted from the client to the host, such as profile *x.pro* transmitted from the host to the client. As such, claim 1 is clearly distinguishable from the approach of Bennett, which is equally unequivocal in its identifying of data objects using a "profile" and not the request. As explained above, Bennett uses a server-stored profile to identify file components and the profile is sent from the server to

the client only in response to the request from the client. The profile request in Bennett thus does not identify any multiple files or data objects. In addition, as shown in Fig. 4, each of the client's subsequent requests asks the server for one component, and hence, at best identifies a single file component, not "a plurality of data objects," as recited in claim 1. See Figs. 4, 6, elem. 266 (requesting components  $x_1, x_2, \dots, x_n$  sequentially).

Moreover, claim 1 is unambiguous about the kind of transfer from the client to the host which must identify the data objects - it must be a request by the client for a data transfer session. Here again, Bennett teaches otherwise, as Fig. 4 of Bennett clearly identifies data objects using transmissions that occur after initiation of the data transfer session between the client and the host. Indeed, the data session in Bennet is established by a transmission of control files from the server to the client (Col. 8:47-50), not a request from the client to the server, as indicated in claim 1.

Thus, Bennett discloses a data transfer protocol with a different request workflow from than the one described in claim 1. At a minimum, Bennett does not teach or suggest "transmitting a request for a data transfer session from a client device to a host, the request identifying a plurality of data objects to be transferred between the client device and the host." Accordingly, Applicants respectfully request reconsideration and withdrawal of the 35 U.S.C. 102(e) rejection of claim 1 and its dependent claims.

Independent claims 10, 17, 37, 45, and 51 similarly describe a request identifying a plurality of data objects and are allowable for the same reasons as explained with respect to claim 1. Accordingly, Applicants respectfully request reconsideration and withdrawal of the rejection of claims 10, 17, 37, 45, and 51 and their dependent claims.

#### **B. 35 U.S.C. 112 Rejection of Claim 1 is Moot**

In the Final Office Action, claim 1 was rejected under 35 U.S.C. § 112, second paragraph because of a missing antecedent for term "plurality of identifiers."

In their response to the Final Office Action, Applicants corrected the missing antecedent issue in claim 1. Applicants' Response of November 9, 2009. The Examiner acknowledged and entered the amendment in the Advisory Action. See Advisory Action of December 7, 2009 at 2. While the Advisory Action does not explicitly withdraw the previous 35 U.S.C. § 112 rejection

Applicant : Kenneth Carbone et al.  
Serial No. : 09/582,297  
Filed : June 3, 2002  
Page : 14 of 26

Attorney's Docket No.: 06975-0029006 / Personlization  
01-DIV3

of claim 1, the rejection is believed to be moot as the antecedent problem has been resolved by the entered amendment. Accordingly, Applicants respectfully request reconsideration and withdrawal of the 35 U.S.C. § 112 rejection of claim 1.

The brief fee of \$540 and a one-month extension of time fee of \$130 is being paid concurrently herewith on the Electronic Filing System (EFS) by way of Deposit Account authorization. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 2/12/10

/Dmitry Brant/  
Dmitry Brant  
Reg. No. 59,133

Fish & Richardson P.C.  
1425 K Street, N.W.  
11th Floor  
Washington, DC 20005-3500  
Telephone: (202) 783-5070  
Facsimile: (877) 769-7945

## **VIII. Claims Appendix**

1. (Previously Presented) A method of asynchronously transferring a plurality of data objects between client and host devices, the method comprising:

transmitting a request for a data transfer session from a client device to a host, the request identifying a plurality of data objects to be transferred between the client device and the host;

in response to the received request, transmitting from the host to the client device a plurality of identifiers for data objects, wherein each identifier is assigned by the host and corresponds to a different one of the data objects to be transferred;

transferring over a network between the host and client devices a data frame that includes an identifier and at least a portion of the corresponding data object;

and repeating the data frame transfers until the plurality of data objects have been transferred.

2. (Previously Presented) The method of claim 1, wherein at least two sequential transfers of a data frame include transferring frames with different identifiers.

3. (Original) The method of claim 1, wherein the transfers of the portions of at least two data objects are interleaved.

4. (Original) The method of claim 1, further comprising: transmitting a data transfer request from the client device to the host device, the transmission of a plurality of identifiers being in response to the data transfer request.

5. (Original) The method of claim 1, wherein the transfers are downloads.

6. (Original) The method of claim 1, wherein a portion of the transfers are uploads and a portion of the transfers are downloads, the uploads and downloads being interleaved.

7. (Previously Presented) The method of claim 1, wherein the transfers of data frames stop at a preselected frame count in the absence of a request for more data frames from a device that receives the data frames.

8. (Original) The method of claim 1, further comprising: transmitting to the client device a size for data frames before the transfers, the data frames transferred being of said size.

9. (Original) The method of claim 1, further comprising: transmitting a frame count to the client device, the frame count corresponding to the number of data frames that the client device can transfer without receiving a request for more data frames.

10. (Previously Presented) A method of asynchronously transferring a plurality of data objects between client and host devices, the method comprising:

transmitting, from a client device to a host, a request for a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host;

in response to the received request, transmitting to a client device a plurality of identifiers and routings of one or more handling processes, wherein each identifier is assigned by the host and corresponds to one of the data objects identified in the request;

transferring between the client and host devices a first data frame that includes a first identifier, a routing of a first handling process, and at least a portion of the data object corresponding to the first identifier;

transferring between the client and host devices a second data frame that includes a second identifier, a routing of a second handling process, and at least a portion of the data object corresponding to the second identifier;

and repeating the data frame transfers until the plurality of data objects have been transferred.



11. (Original) The method of claim 10, further comprising: writing the portions of the data objects to first and second storage locations to which the respective first and second identifiers are assigned.

12. (Original) The method of claim 11, wherein the writes of the first and second portions of the data objects corresponding to the first and second identifiers are controlled by the first and second handling processes, respectively.

13. (Original) The method of claim 10, wherein the first and second handling processes handle uploads of data objects for first and second data objects, respectively.

14. (Original) The method of claim 13, wherein the first and second data objects include data for first and second images, respectively.

15. (Original) The method of claim 10, wherein the transfers of data frames including the first identifier stop at a preselected frame count in the absence of a request for more data frames from a device that receives the data frames.

16. (Original) The method of claim 10, wherein the request for more data frames includes the routing of the first handling process.

17. (Previously Presented) A method of asynchronously transferring data between host and client devices, comprising:

receiving from a client device a request a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host;

sending to the client device a frame defining a session protocol that assigns an identifier to each data object identified in the request, wherein each identifier is assigned by the host; and

transferring a plurality of data frames between the client and host devices, each data frame comprising a data portion of a data object and an identifier assigned to the data object including said data portion.

18. (Original) The method of claim 17, wherein the transferring of data frames includes a data upload.

19. (Original) The method of claim 18, further comprising: writing a particular data portion to a storage volume assigned to a particular identifier in response to receiving a data frame including the particular identifier and data portion, unique data objects being assigned to each storage volume.

20. (Original) The method of claim 17, further comprising: receiving a second frame from the client device requesting a second data transfer session;

    sending a second frame to the client device defining a second session protocol that assigns an identifier to each data object of the second session;

    transferring a plurality of second data frames between the client and host devices, each second data frame including a second data portion and an identifier assigned to a data object including the second data portion.

21. (Original) The method of claim 20, wherein the transfers of first and second data frames are interleaved.

22. (Original) The method of claim 20, wherein the transfers of second data frames are downloads from the host device.

23. (Original) The method of claim 17, further comprising:  
    receiving a frame from a second client device requesting a second data transfer session;  
    sending a frame to the second client device defining a second session protocol that assigns an identifier to each second data object of the second session; and

    transferring a plurality of second data frames between the second client and host devices, each second data frame including a second data portion of a second data object and an associated identifier.

24. (Original) The method of claim 17, further comprising: 1 sending to the client device a routing for a handling program assigned to each data object;  
and wherein each data frame includes the routing of the handling program assigned to the data object therein.

25. (Original) The method of claim 24, wherein fist and second data objects are assigned first and second handling programs, respectively.

26. (Original) The method of claim 24, further comprising: writing a particular data portion to a storage volume assigned to a particular identifier in response to receiving a data frame including the particular identifier and data portion, unique data objects being assigned to each storage volume.

27. (Original) The method of claim 26, further comprising: controlling the write with the handling program assigned to the data object being written.

28-36. (Cancelled)

37. (Previously Presented) A computer readable medium encoding a program of instructions to asynchronously transfer a plurality of data objects between client and host devices, the instructions comprising:

transmit, from a client device to a host, a request for a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host;

transmit to a client device the plurality of identifiers for data objects, wherein each identifier is assigned by the host and corresponds to a different one of the data objects to be transferred;

transfer over a network between the host and client devices a data frame that includes an identifier and at least a portion of the corresponding data object; and

repeat the data frame transfers until the plurality of data objects have been transferred.

38. (Original) The computer readable medium of claim 37, wherein at least two sequential transfers of a data frame include transfers of frames with different identifiers.

39. (Original) The computer readable medium of claim 37, wherein the transfers of the portions of at least two data objects are interleaved.

40. (Original) The computer readable medium of claim 37, the instructions further comprising: transmit a data transfer request from the client device to the host device, the transmission of a plurality of identifiers being in response to the data transfer request.

41. (Original) The computer readable medium of claim 37, wherein the transfers are downloads.

42. (Original) The computer readable medium of claim 37, wherein a portion of the transfers are uploads and a portion of the transfers are downloads, the uploads and downloads being interleaved.

43. (Original) The computer readable medium of claim 37, wherein the transfers of data frames stop at a preselected frame count in the absence of a request for more data frames from a device that receives the data frames.

44. (Original) The computer readable medium of claim 37, the instructions further comprising: transmit a frame count to the client device, the frame count corresponding to the number of data frames that the client device can transfer without receiving a request for more data frames.

45. (Previously Presented) A computer readable medium encoding a program of instructions to asynchronously transfer a plurality of data objects between client and host devices, the instructions comprising:

transmitting, from a client device to a host, a request for a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host;

transmit to the client device a plurality of identifiers and routings of one or more handling processes, wherein each identifier is assigned by the host and corresponds to one of the data objects identified in the request;

transfer between the client and host devices a first data frame that includes a first identifier, a routing of a first handling process, and at least a portion of the data object corresponding to the first identifier;

transfer between the client and host devices a second data frame that includes a second identifier, a routing of a second handling process, and at least a portion of the data object corresponding to the second identifier; and

repeat the data frame transfers until the plurality of data objects have been transferred.

46. (Original) The computer readable medium of claim 45, the instructions further comprising: write the portions of the data objects to first and second storage locations to which the respective first and second identifiers are assigned.

47. (Original) The computer readable medium of claim 46, wherein the writes of the first and second portions of the data objects corresponding to the first and second identifiers are controlled by the first and second handling processes, respectively.

48. (Original) The computer readable medium of claim 45, wherein the first and second processes handle uploads of files for first and second data objects.

49. (Original) The computer readable medium of claim 48, wherein the first and second data objects are data for first and second images, respectively.

50. (Original) The computer readable medium of claim 45, wherein the transfers of data frames including the first identifier stop at a preselected frame count in the absence of a request for more data frames from a device that receives the data frames.

51. (Previously Presented) A computer readable medium encoding a program of instructions to asynchronously transferring data between host and client devices, the instructions comprising: receive from a client device a request for a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host;

send to the client device a frame defining a session protocol that assigns an identifier to each data object identified in the request; wherein each identifier is assigned by the host ; and

transfer a plurality of data frames between the client and host devices, each data frame comprising a data portion and an identifier assigned to a data object including said data portion.

52. (Previously Presented) The computer readable medium of claim 51, wherein the transfer of data frames includes a data upload to the host device.

53. (Original) The computer readable medium of claim 52, the instructions further comprising: write a particular data portion to a storage volume assigned to a particular identifier in response to receiving a data frame including the particular identifier and data portion, unique data objects being assigned to each storage volume.

54. (Original) The computer readable medium of claim 51, the instructions further comprising: receive a second frame from the client device requesting a second data transfer session;

send a second frame to the client device defining a second session protocol that assigns an identifier to each data object of the second session;

transfer a plurality of second data frames between the client and host devices, each second data frame including a second data portion and an identifier assigned to a data object including the second data portion.

55. (Original) The computer readable medium of claim 54, wherein the transfer of first and second data frames are interleaved.

56. (Original) The computer readable medium of claim 54, wherein the transfers of second data frames are downloads from the host device.

57. (Original) The computer readable medium of claim 51, the instructions further comprising: receive a frame from a second client device requesting a second data transfer session;

send a frame to the second client device defining a second session protocol that assigns an identifier to each data object of the second session; and

transfer a plurality of second data frames between the second client and host devices, each second data message including a data portion of a second data object and an identifier assigned to the second data object.

58. (Original) The computer readable medium of claim 51, the instructions further comprising: send to the client device a routing for a handling program of assigned to each data object;

and wherein each data frame includes the routing of the handling program assigned to the data object therein.

59. (Original) The computer readable medium of claim 58, wherein first and second data objects are assigned first and second handling programs, respectively.

60. (Previously Presented) The computer readable medium of claim 58, the instructions further comprising: write a particular data portion to a storage volume assigned to a particular identifier in response to receiving a data frame including the particular identifier and data portion, unique data objects being assigned to each storage volume;

Applicant : Kenneth Carbone et al.  
Serial No. : 09/582,297  
Filed : June 3, 2002  
Page : 24 of 26

Attorney's Docket No.: 06975-0029006 / Personlization  
01-DIV3

and control the write with the handling program assigned to the data object being written.

61-69. (Cancelled)



Applicant : Kenneth Carbone et al.  
Serial No. : 09/582,297  
Filed : June 3, 2002  
Page : 25 of 26

Attorney's Docket No.: 06975-0029006 / Personlization  
01-DIV3

**IX. Evidence Appendix**

None.

Applicant : Kenneth Carbone et al.  
Serial No. : 09/582,297  
Filed : June 3, 2002  
Page : 26 of 26

Attorney's Docket No.: 06975-0029006 / Personlization  
01-DIV3

**X. Related Proceedings Appendix**

None.